



Secureki - Cisco Duo & Webex Teams Integration



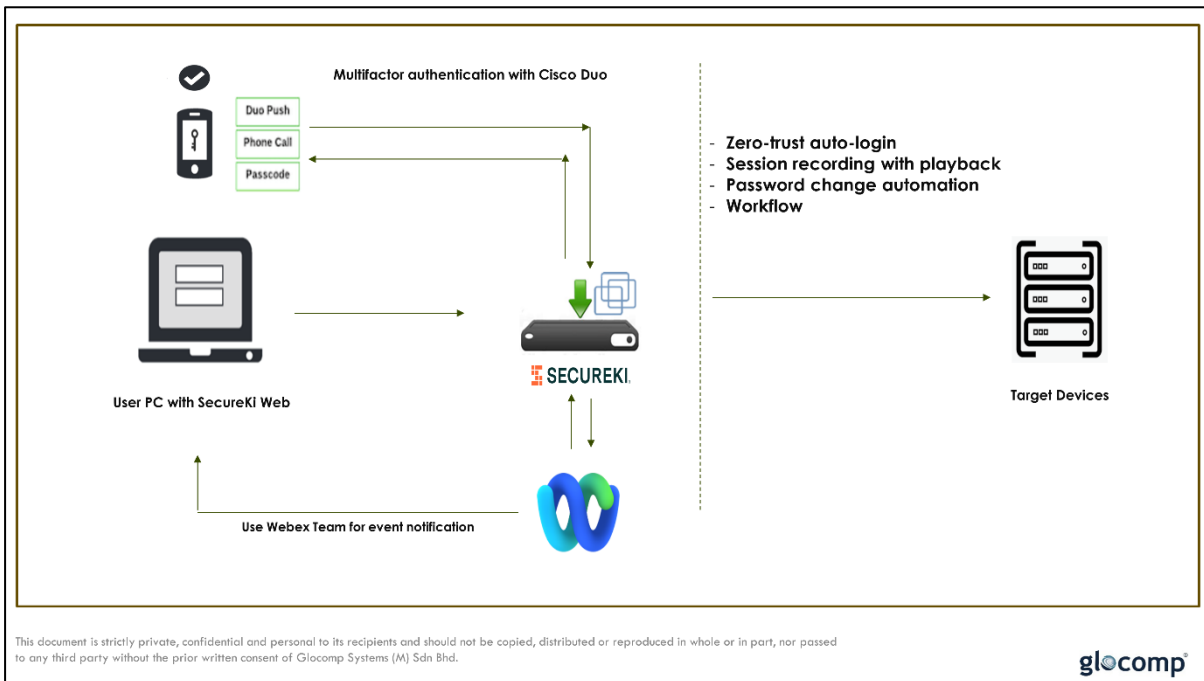
Technical Specifications

Company Name : Glocomp Systems (M) Sdn Bhd
Application Name : SecureKi

Contents

Overview	3
Use Case 1 – User approve the DUO push.....	4
1.1. APPM WEB.....	4
Use Case 2 – User denies the DUO push.....	8
2.1. APPM WEB.....	8
Use Case 3 – Not Valid Cisco Duo User	11
3.1. APPM WEB.....	11
Use Case 4 – Cisco DUO is inaccessible.....	13
4.1. APPM WEB.....	13
Use Case 5 – User does not response to the DUO Push / Timed Out	15
5.1. APPM WEB.....	15
Use Case 6 – User enters ‘6-digits passcode’ instead of ‘push’	18
6.1. APPM WEB.....	18
Use Case 7 – Notification in Webex Team via Webex API	21
7.1. APPM SERVICE.....	21
7.2. Event Notification from Webex Team Space.....	22
Use Case 8 – Approval Notification via WebEx Cards and Buttons	25
8.1. APPM SERVICE.....	25
Use Case 9 – Workflow Approval via WebEx Team Space	26
9.1. APPM SERVICE.....	26

Overview



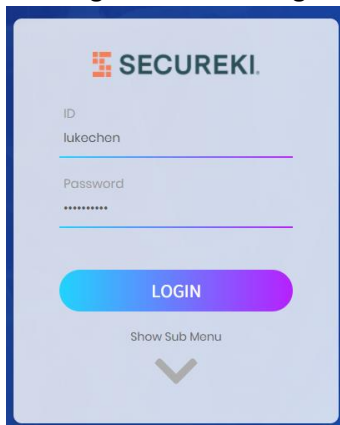
API Steps:-

1. Check if DUO is active (auth/v2/ping)
2. Send PUSH notification to **Cisco DUO** users
 - a. Encode the password (generate from *date_authorization_header.py*)
 - b. Call the POST api (auth/v2/auth)
3. Wait for the API to response.
4. Success/Fail to login to APPM
5. User access to SSH/RDP using APPMClient
6. Admin will receive event notification via **WebexTeam**

Use Case 1 – User approve the DUO push

1.1. APPM WEB

1. User login to APPM using username and password

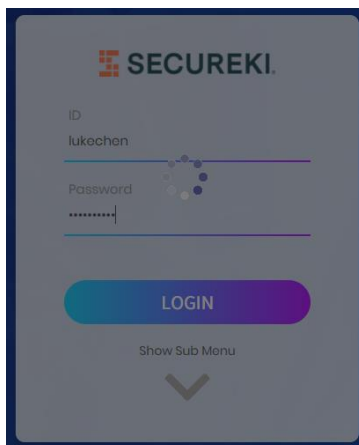


The image shows a login form for 'SECUREKI'. It has a light blue background with a dark blue border. At the top is the 'SECUREKI.' logo. Below it are two input fields: 'ID' with the value 'lukechen' and 'Password' with masked characters '*****'. A large, rounded 'LOGIN' button is in the center. Below the button is a 'Show Sub Menu' link with a downward arrow.

2. After user clicks the Login button, APPM should check if DUO server is up and running.

*If stat is **OK**, proceed to step 3.*

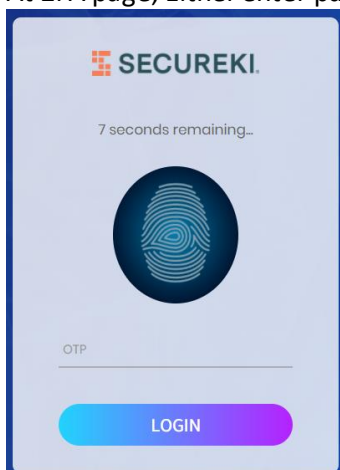
*If stat is **Fail**, show message "There was a problem accessing to DUO"*



The image shows the same login form as before, but with a loading spinner (a circle of dots) over the password field, indicating a processing or loading state.

GET: `https://{{duo-API-HOST}}/auth/v2/ping`
 {"response": {"time": 1619186110}, "stat": "OK"}

3. At 2FA page, Either enter passcode or approve/reject DUO Push



The image shows a 2FA (Two-Factor Authentication) page for 'SECUREKI'. It has a light blue background with a dark blue border. At the top is the 'SECUREKI.' logo. Below it is a timer that says '7 seconds remaining...'. In the center is a large circular graphic with a fingerprint icon. Below this is an 'OTP' input field. At the bottom is a large, rounded 'LOGIN' button.

Authorization

Key	Value	Remarks
Username	<To be provided>	Client ID
Password	<<encrypted string>>	Generate from <i>date_authorization_header.py</i> Format as below:- <Date> <GET/POST> <API_ADDRESS> <API_URL> <API_QUERY> e.g. Fri, 23 Apr 2021 23:59:03 +0800 POST api-*.duosecurity.com /auth/v2/auth device=auto&factor=push&username=lukechen

HEADER

Key	Value	Remarks
Date	Fri, 23 Apr 2021 23:59:03 +0800	Date format as RFC2822
Content-Type	application/x-www-form-urlencoded	

PUSH

Key	Value	Remarks
device	auto	Auto select user's device to be pushed
factor	push / auto	Get this value from Duo Radius Code If user key in "push", set factor=push
username	Secureki	Person ID

POST: <https://{{duo-API-HOST}}/auth/v2/auth?device=auto&factor=push&username=lukechen>

PASSCODE

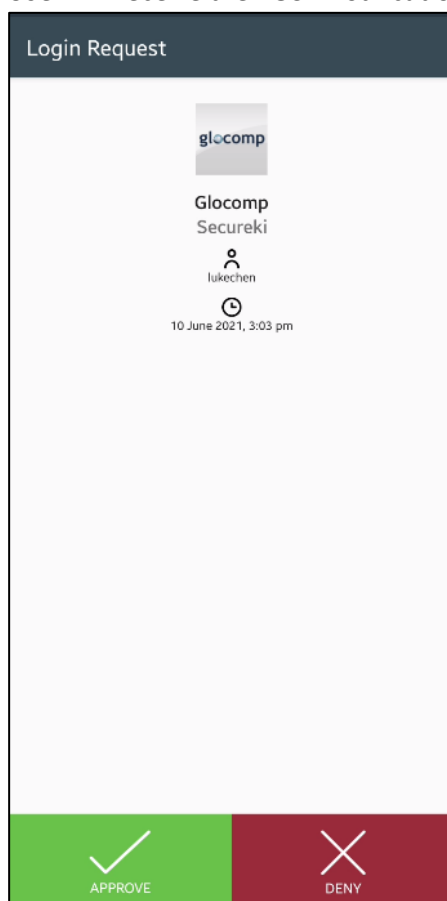
Key	Value	Remarks
factor	passcode	Get this value from Duo Radius Code If user key in 6-digits number, set factor=passcode
Passcode	123456	6-digits passcode
username	Secureki	Person ID

POST: <https://{{duo-API-HOST}}/auth/v2/auth?factor=passcode&passcode=123456&username=lukechen>

Response/Return:-

Status	Results
Approve	<code>{"response": {"result": "allow", "status": "allow", "status_msg": "Success. Logging you in..."}, "stat": "OK"}</code>
Deny	<code>{"response": {"result": "deny", "status": "deny", "status_msg": "Login request denied."}, "stat": "OK"}</code> <code>{"response": {"result": "deny", "status": "fraud", "status_msg": "Login request reported as fraudulent."}, "stat": "OK"}</code>
Timed out	<code>{"response": {"result": "deny", "status": "timeout", "status_msg": "Login timed out."}, "stat": "OK"}</code>
Invalid User	<code>{"code": 40002, "message": "Invalid request parameters", "message_detail": "username", "stat": "FAIL"}</code>

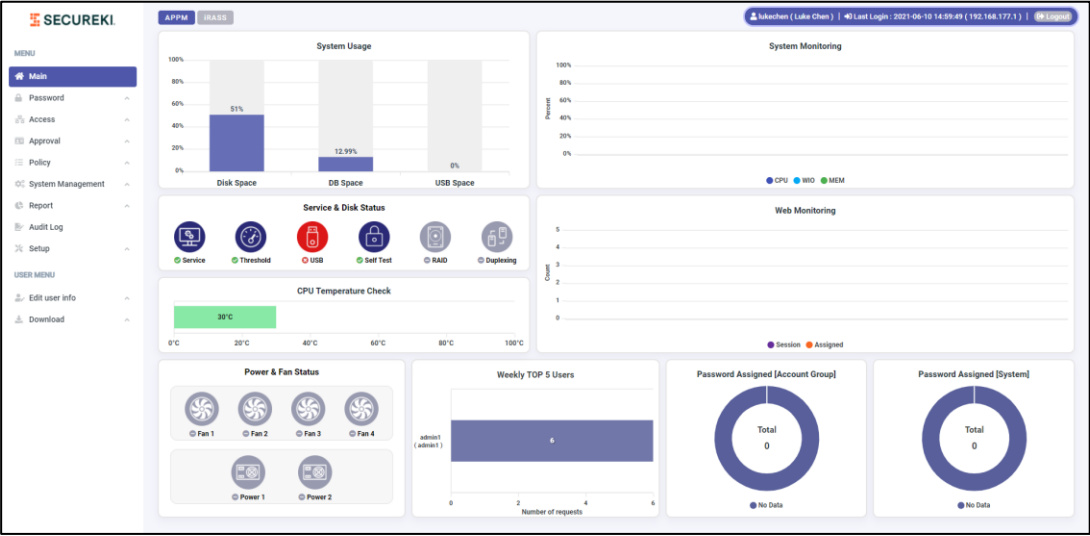
4. User will receive the **PUSH** notification to approve or deny



(Phone notification)
(duo mobile app)

5. APPM Web will wait for the user to response on their DUO mobile (**Timeout: 1 min**)

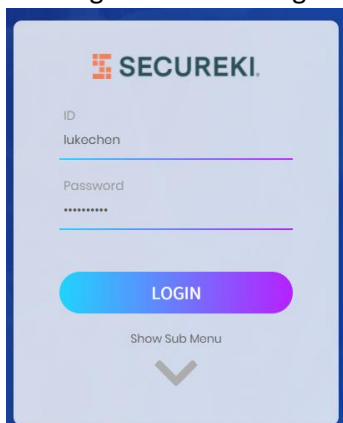
6. Once approve, user successfully login to APPM Web



Use Case 2 – User denies the DUO push

2.1. APPM WEB

1. User login to APPM using username and password

A screenshot of the SecureKI login interface. At the top is the 'SECUREKI.' logo. Below it are two input fields: 'ID' with the value 'lukechen' and 'Password' with masked characters. A blue 'LOGIN' button is centered below the fields. At the bottom, there is a 'Show Sub Menu' link and a downward-pointing chevron icon.

2. After user clicks the Login button, APPM should check if DUO server is up and running.

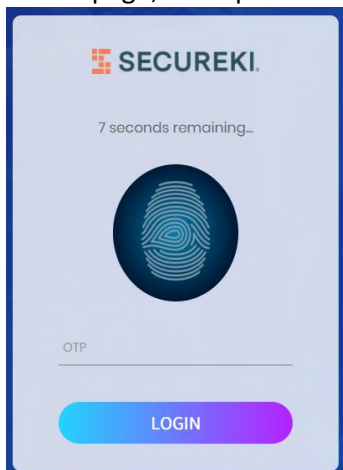
*If stat is **OK**, proceed to step 3.*

*If stat is **Fail**, show message "There was a problem accessing to DUO"*

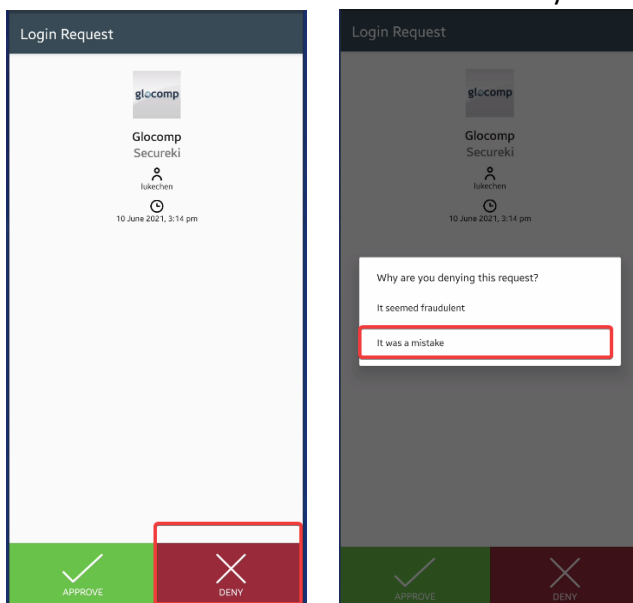
A screenshot of the SecureKI login interface, identical to the previous one, but with a semi-transparent grey overlay. In the center of the overlay is a circular icon containing a network diagram with nodes and connecting lines, indicating a connection or status check.

GET: `https://{{duo-API-HOST}}/auth/v2/ping`
{ "response": { "time": 1619186110, "stat": "OK" }

3. At 2FA page, Enter passcode or approve/deny the DUO PUSH

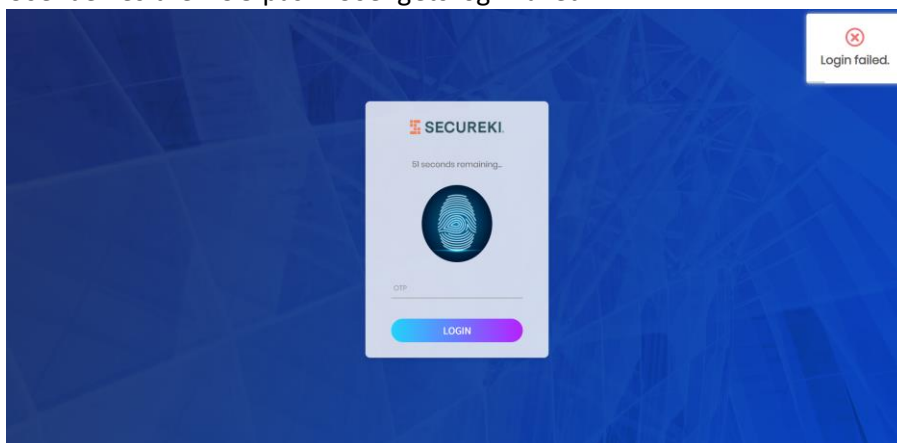
A screenshot of the SecureKI 2FA (Two-Factor Authentication) screen. At the top is the 'SECUREKI.' logo. Below it, a timer shows '7 seconds remaining...'. In the center is a large circular icon with a fingerprint. Below the icon is an 'OTP' input field. At the bottom is a blue 'LOGIN' button.

4. User will receive the **PUSH** notification to deny it



(Phone notification)
(duo mobile app)

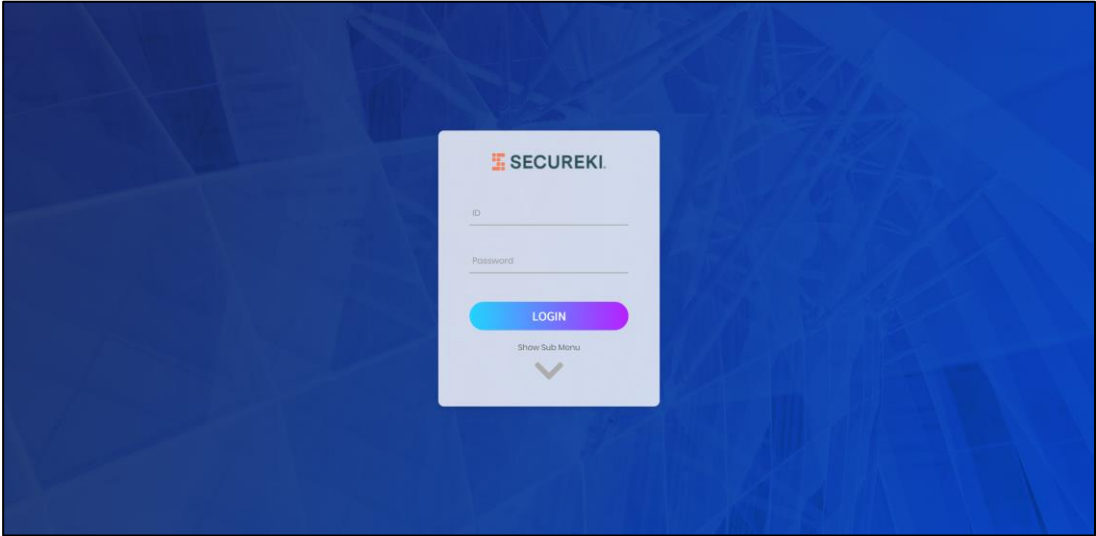
5. APPM Web will wait for the user to response on their DUO mobile (**Timeout: 1 min**)
6. User denies the DUO push. User gets login failed.



Response/Return:-

Status	Results
Deny	<pre>{ "response": { "result": "deny", "status": "deny", "status_msg": "Login request denied.", "stat": "OK" } }</pre> <pre>{ "response": { "result": "deny", "status": "fraud", "status_msg": "Login request reported as fraudulent.", "stat": "OK" } }</pre>

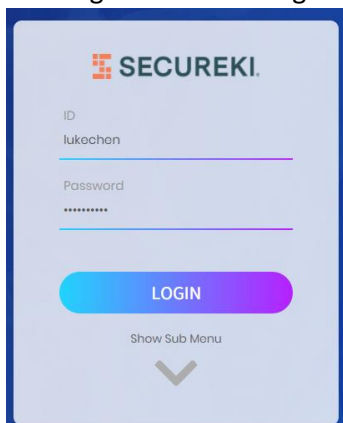
7. Redirect the APPM back to Login page



Use Case 3 – Not Valid Cisco Duo User

3.1. APPM WEB

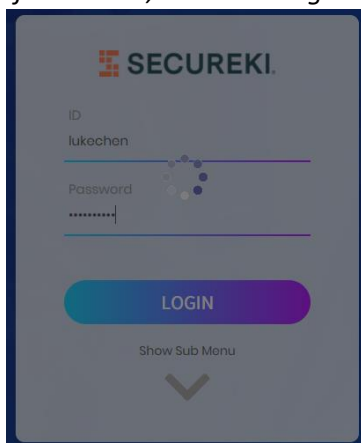
1. User login to APPM using username and password

A login form for SecureKI. It has a header with the SecureKI logo. Below the logo are two input fields: 'ID' with the value 'lukechen' and 'Password' with masked characters. A blue 'LOGIN' button is centered below the fields. At the bottom, there is a 'Show Sub Menu' link with a downward arrow.

2. After user clicks the Login button, APPM should check if DUO server is up and running.

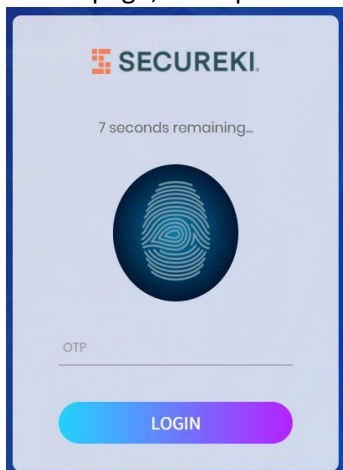
*If stat is **OK**, proceed to step 3.*

*If stat is **Fail**, show message "There was a problem accessing to DUO"*

The same login form as before, but with a loading spinner (a circle of dots) over the password field, indicating a processing or loading state.

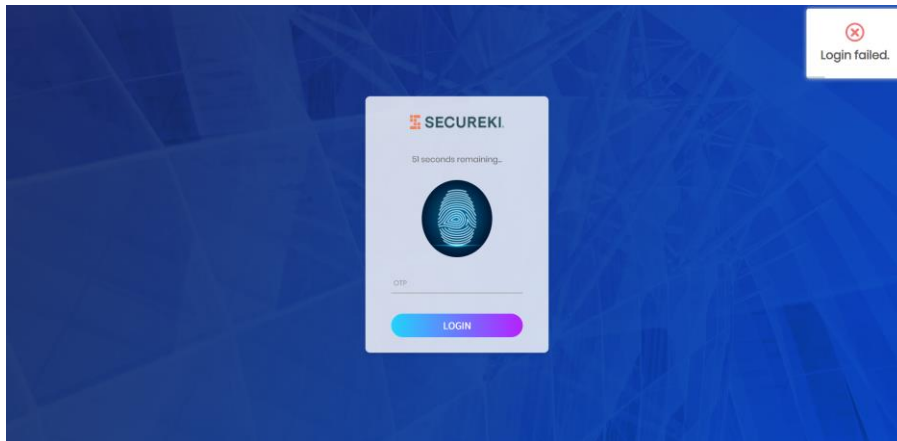
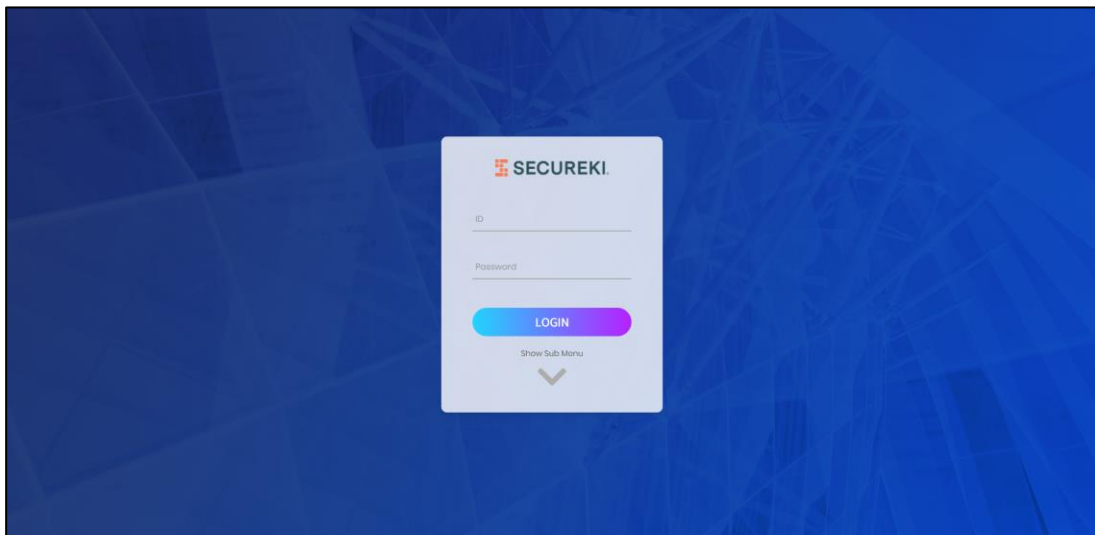
GET: `https://{{duo-API-HOST}}/auth/v2/ping`
{ "response": { "time": 1619186110, "stat": "OK" } }

3. At 2FA page, Enter passcode or approve/deny the DUO PUSH

A 2FA (Two-Factor Authentication) page for SecureKI. It shows a timer '7 seconds remaining...'. Below the timer is a circular graphic with a fingerprint icon. At the bottom, there is an 'OTP' input field and a blue 'LOGIN' button.

Response/Return:-

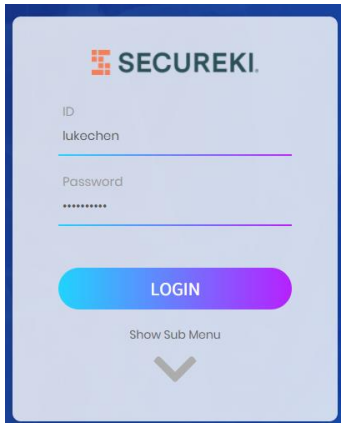
Status	Results
Invalid User	{"code": 40002, "message": "Invalid request parameters", "message_detail": "username", "stat": "FAIL"}

4. Error prompt “Login Failed.”**5. Redirect the user back to main login page.**

Use Case 4 – Cisco DUO is inaccessible

4.1. APPM WEB

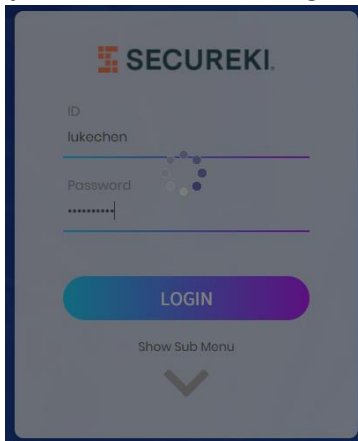
1. User login to APPM using username and password



2. After user clicks the Login button, APPM should check if DUO server is up and running.

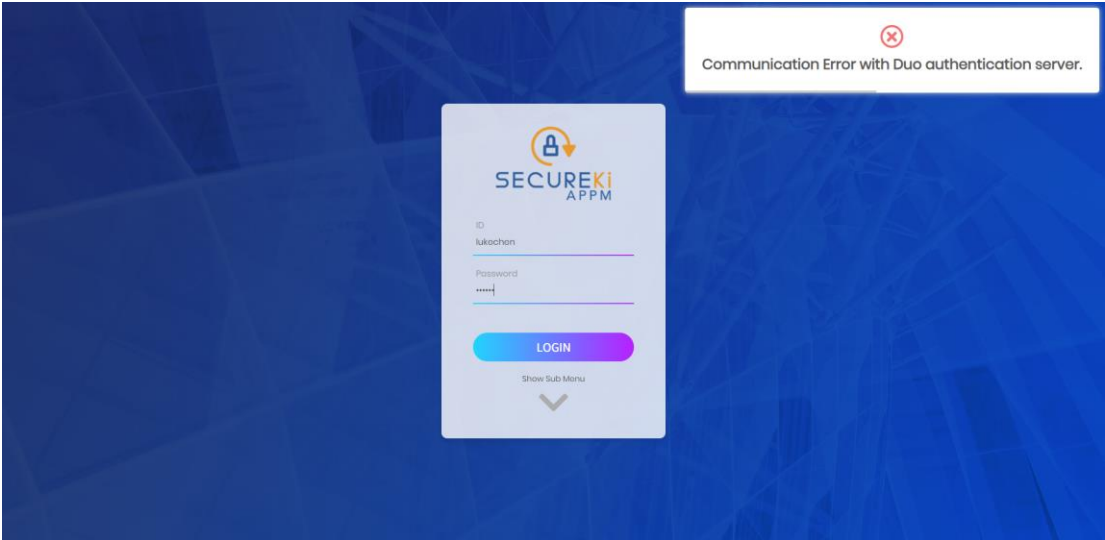
*If stat is **OK**, proceed to step 3.*

*If stat is **Fail**, show message "There was a problem accessing to DUO"*



GET: `https://{{duo-API-HOST}}/auth/v2/ping`
{ "response": { "time": 1619186110, "stat": "Fail" } }

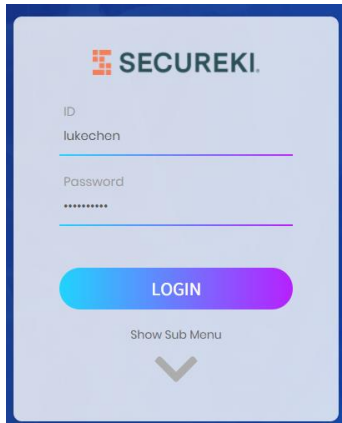
3. APPM login failed when ping is fail. (e.g. There was an error communicating with Duo authentication server.)



Use Case 5 – User does not response to the DUO Push / Timed Out

5.1. APPM WEB

1. User login to APPM using username and password

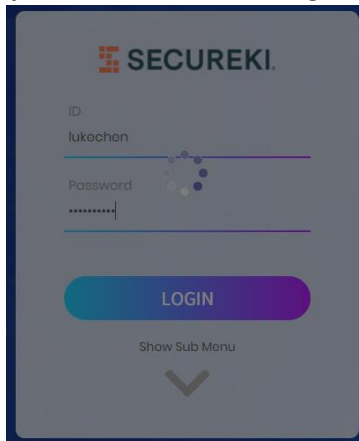


The image shows the Secureki login interface. At the top is the 'SECUREKI.' logo. Below it are two input fields: 'ID' with the value 'lukechen' and 'Password' with masked characters. A blue 'LOGIN' button is centered below the fields. At the bottom, there is a 'Show Sub Menu' link with a downward arrow icon.

2. After user clicks the Login button, APPM should check if DUO server is up and running.

*If stat is **OK**, proceed to step 3.*

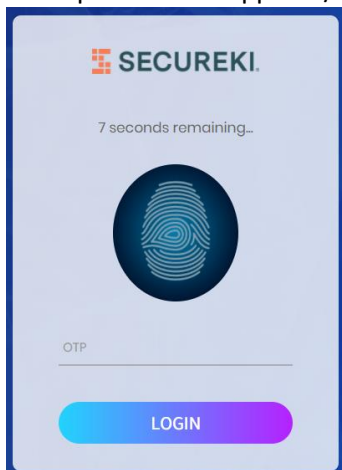
*If stat is **Fail**, show message "There was a problem accessing to DUO"*



The image shows the Secureki login interface with a loading spinner (a circle of dots) over the password field, indicating a processing or loading state.

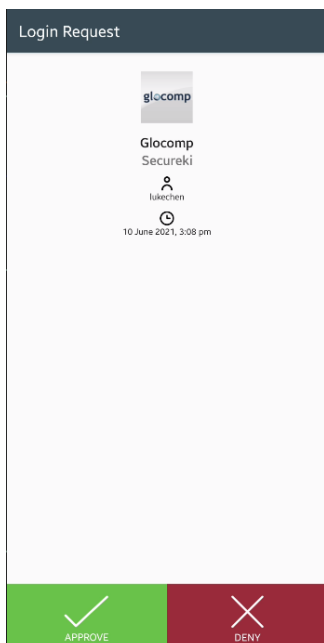
GET: `https://{{duo-API-HOST}}/auth/v2/ping`
 {"response": {"time": 1619186110, "stat": "OK"}}

3. Enter passcode or approve/deny the DUO PUSH



The image shows the Secureki login interface during a DUO push notification. At the top is the 'SECUREKI.' logo. Below it, a timer says '7 seconds remaining...'. In the center is a large circular icon with a fingerprint. Below that is an 'OTP' input field. At the bottom is a blue 'LOGIN' button.

- User will receive the **PUSH** notification to approve or deny



- APPM Web will wait for the user to response on their DUO mobile (**Timeout: 1 min**)

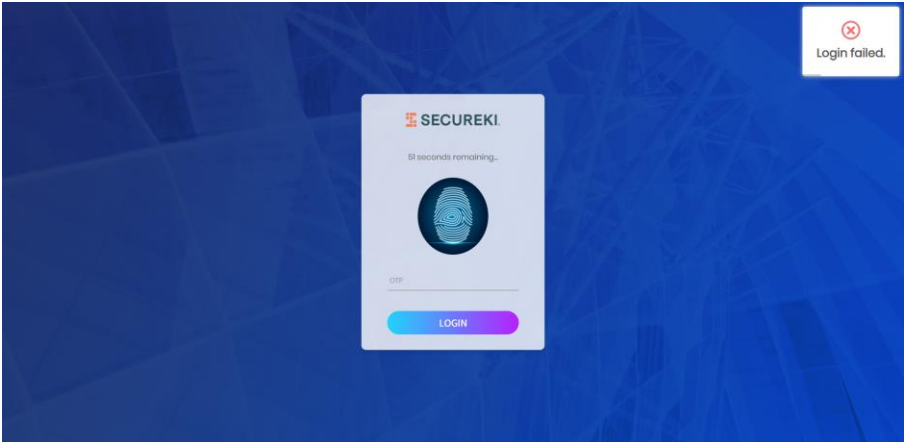


- User does NOT approve/deny the request.
- Request is timed out after 1 minute.

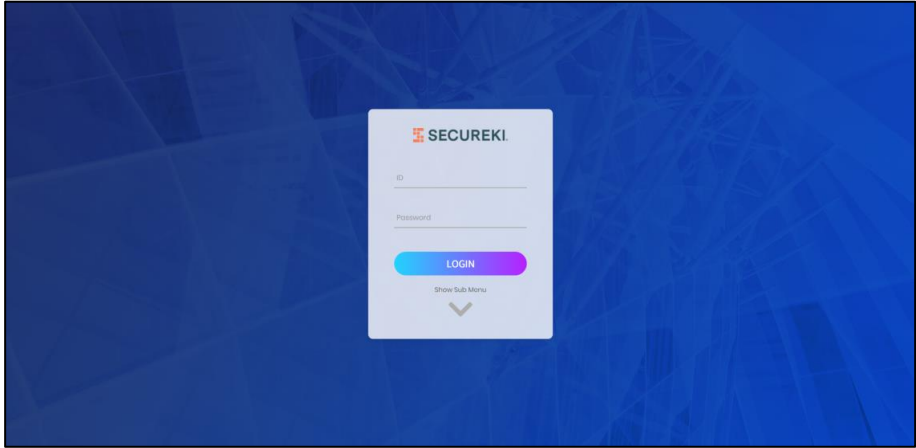
Response/Return:-

Status	Results
Timed out	<code>{"response":{"result":"deny","status":"timeout","status_msg":"Login timed out."},"stat":"OK"}</code>

- Error prompt using the status_msg (e.g. **Login timed out.**)



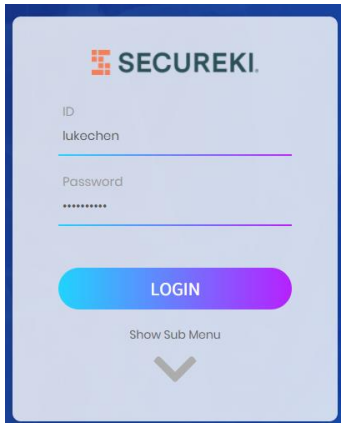
9. Redirect the APPM back to Login page



Use Case 6 – User enters ‘6-digits passcode’ instead of ‘push’

6.1. APPM WEB

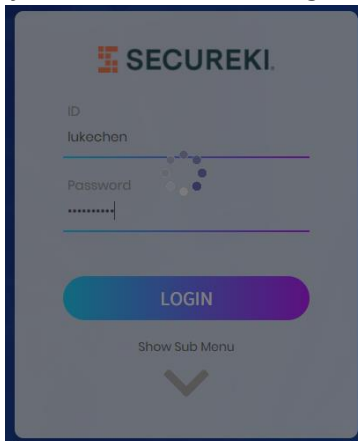
1. User login to APPM using username and password



2. After user clicks the Login button, APPM should check if DUO server is up and running.

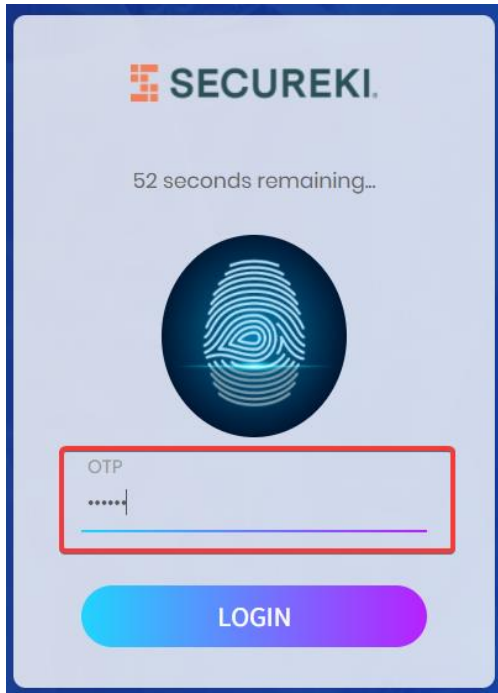
*If stat is **OK**, proceed to step 3.*

*If stat is **Fail**, show message “There was a problem accessing to DUO”*



GET: `https://{{duo-API-HOST}}/auth/v2/ping`
{ "response": { "time": 1619186110, "stat": "OK" }

3. User will type “6-digits passcode” and click Login



Authorization

Key	Value	Remarks
Username	<To be provided>	Client ID
Password	<<encrypted string>>	<p>Generate from <i>date_authorization_header.py</i></p> <p>Format as below:-</p> <p><Date> <GET/POST> <API_ADDRESS> <API_URL> <API_QUERY></p> <p>e.g. Fri, 23 Apr 2021 23:59:03 +0800 POST api-*.duosecurity.com /auth/v2/auth factor=passcode&passcode=123456&username=lukechen</p>

HEADER

Key	Value	Remarks
Date	Fri, 23 Apr 2021 23:59:03 +0800	Date format as RFC2822
Content-Type	application/x-www-form-urlencoded	

PASSCODE

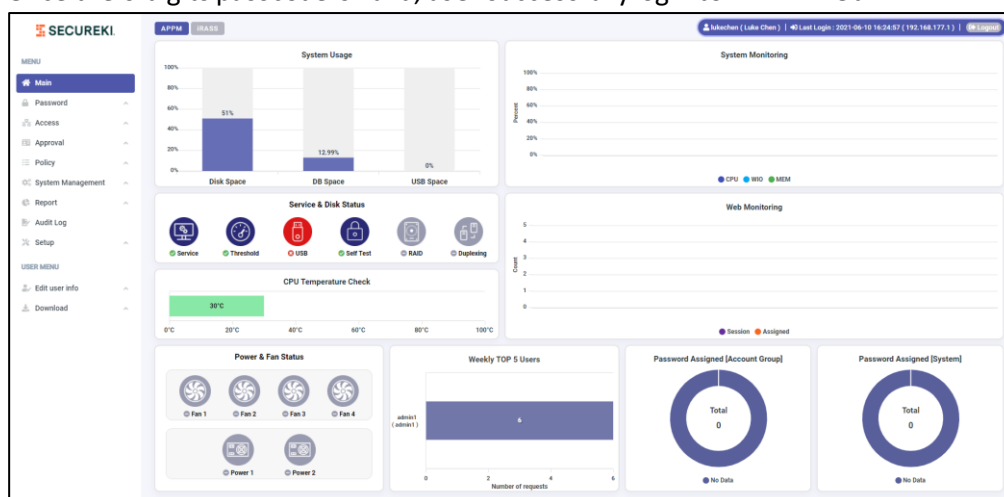
Key	Value	Remarks
factor	passcode	Get this value from Duo Radius Code If user key in 6-digits number, set factor=passcode
Passcode	123456	6-digits passcode
username	Secureki	Person ID

POST: <https://{{duo-API-HOST}}/auth/v2/auth?factor=passcode&passcode=123456&username=lukechen>

Response/Return:-

Status	Results
Approve	<code>{"response":{"result":"allow","status":"allow","status_msg":"Success. Logging you in..."},"stat":"OK"}</code>
Deny	<code>{"response":{"result":"deny","status":"deny","status_msg":"Login request denied."},"stat":"OK"}</code> <code>{"response":{"result":"deny","status":"fraud","status_msg":"Login request reported as fraudulent."},"stat":"OK"}</code>
Timed out	<code>{"response":{"result":"deny","status":"timeout","status_msg":"Login timed out."},"stat":"OK"}</code>
Invalid User	<code>{"code":40002,"message":"Invalid request parameters","message_detail":"username","stat":"FAIL"}</code>

- APPM Web will wait for the user to response on their DUO mobile (**Timeout: 1 min**)
- Once the 6-digits passcode is valid, user successfully login to APPM Web



Use Case 7 – Notification in Webex Team via Webex API

7.1. APPM SERVICE

1. APPM Service will constantly check database for events. Once an event matched the criteria it will send a message to pre-created Webex Space via code snippet below

```
curl --request POST --url https://webexapis.com/v1/messages --header 'Authorization: Bearer ACCESS_TOKEN' --header 'Content-Type: application/json' --data '{"roomId": "ROOM_ID", "text": "FORMATTED_MESSAGE"}
```

Key	Value	Remarks
ACCESS_TOKEN	<access token>	Token is generated when creating bot at https://developer.webex.com/
ROOM_ID	<room id>	Room id retrieved via https://webexapis.com/v1/rooms
FORMATTED_MESSAGE	<pre-formatted message>	

HEADER

Key	Value	Remarks
Content-Type	application/json	

RESPONSE

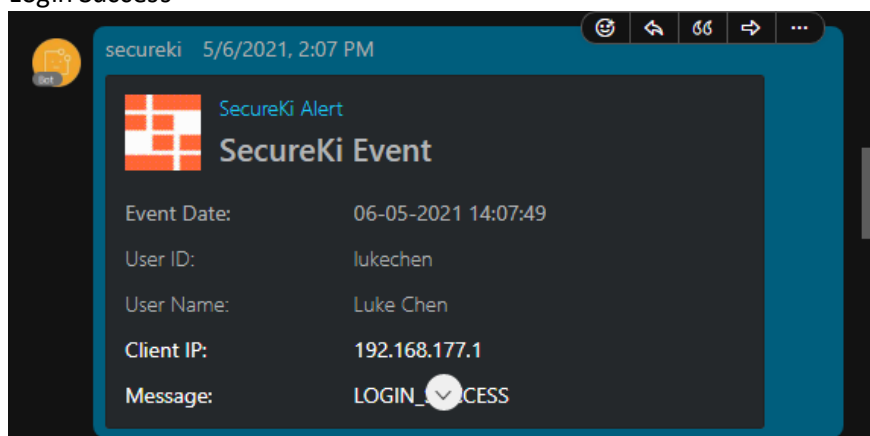
below are the response codes refer to webexapi documentation

Code	Status	Description
200	OK	Successful request with body content.
204	No Content	Successful request without body content.
400	Bad Request	The request was invalid or cannot be otherwise served. An accompanying error message will explain further.
401	Unauthorized	Authentication credentials were missing or incorrect.
403	Forbidden	The request is understood, but it has been refused or access is not allowed.
404	Not Found	The URI requested is invalid or the resource requested, such as a user, does not exist. Also returned when the requested format is not supported by the requested method.
405	Method Not Allowed	The request was made to a resource using an HTTP request method that is not supported.
409	Conflict	The request could not be processed because it conflicts with some established rule of the system. For example, a person may not be added to a room more than once.
410	Gone	The requested resource is no longer available.

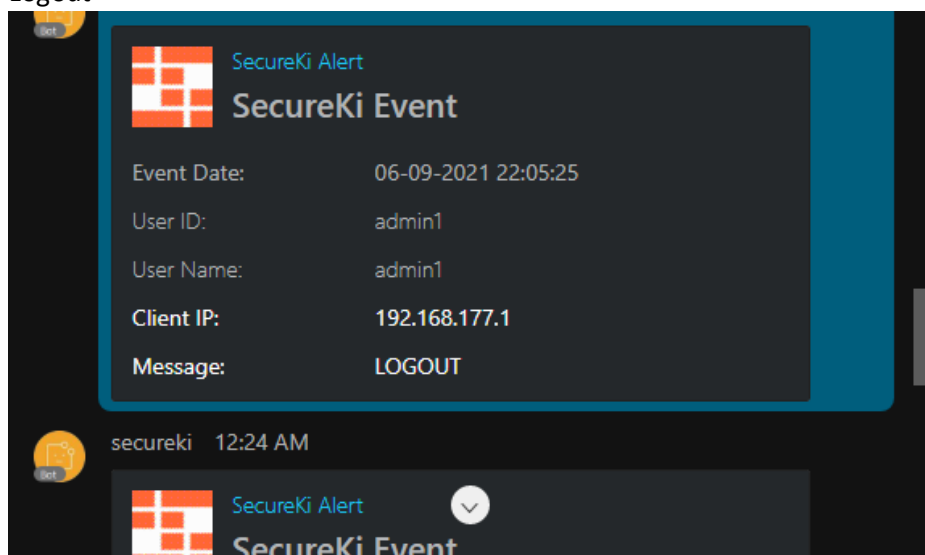
Code	Status	Description
415	Unsupported Media Type	The request was made to a resource without specifying a media type or used a media type that is not supported.
423	Locked	The requested resource is temporarily unavailable. A Retry-After header may be present that specifies how many seconds you need to wait before attempting the request again.
428	Precondition Required	File(s) cannot be scanned for malware and need to be force downloaded.
429	Too Many Requests	Too many requests have been sent in a given amount of time and the request has been rate limited. A Retry-After header should be present that specifies how many seconds you need to wait before a successful request can be made.
500	Internal Server Error	Something went wrong on the server. If the issue persists, feel free to contact the Webex Developer Support team .
502	Bad Gateway	The server received an invalid response from an upstream server while processing the request. Try again later.
503	Service Unavailable	Server is overloaded with requests. Try again later.
504	Gateway Timeout	An upstream server failed to respond on time. If your query uses max parameter, please try to reduce it.

7.2. Event Notification from Webex Team Space

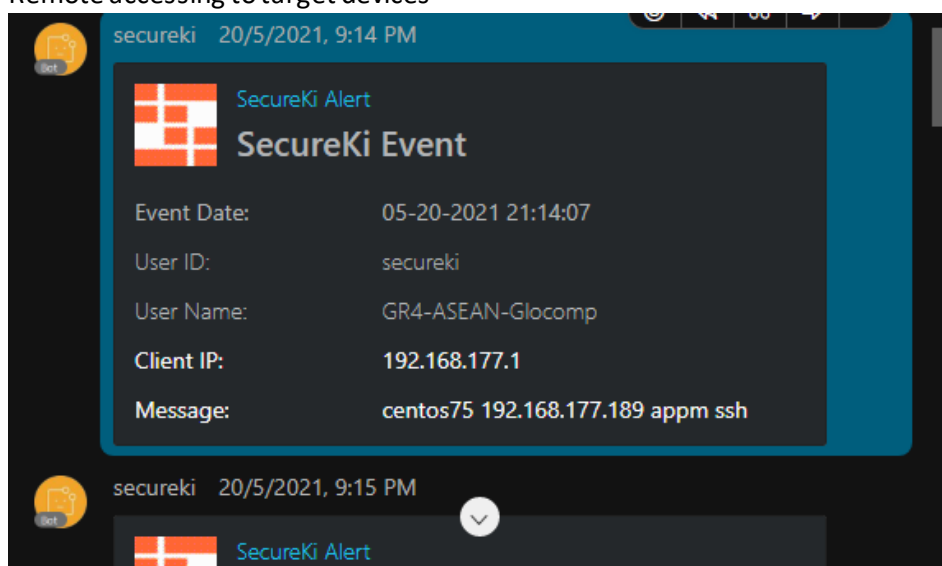
1. Login Success



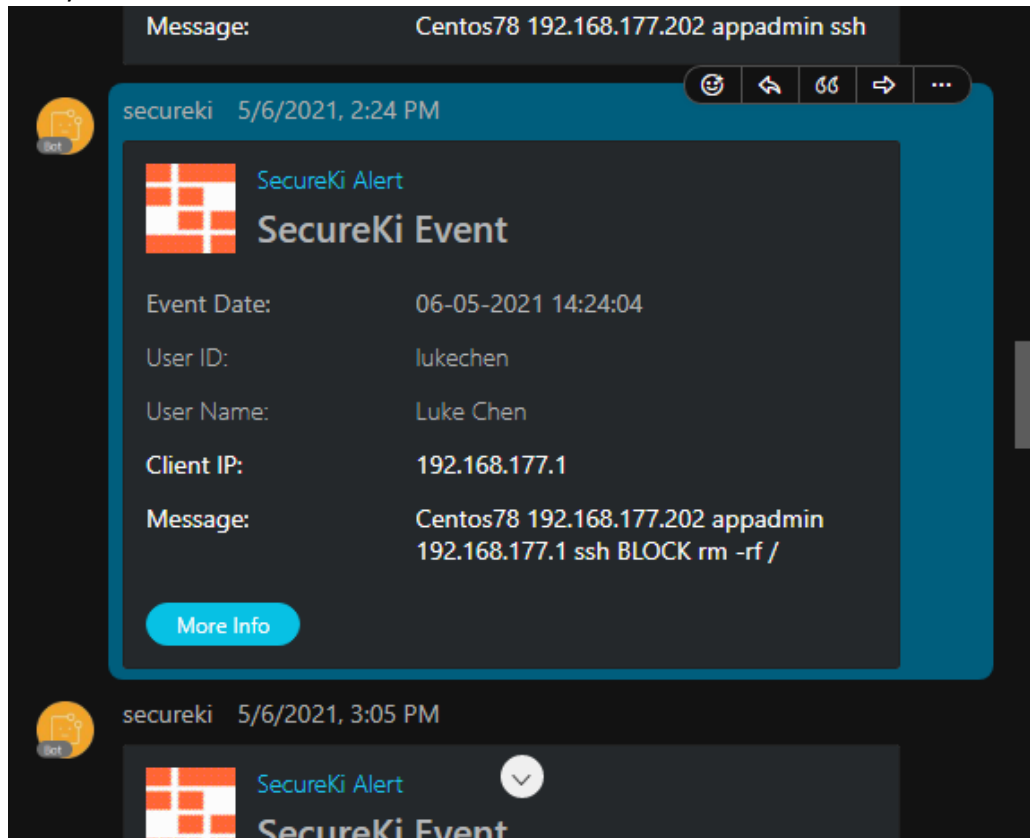
2. Logout



3. Remote accessing to target devices



4. Policy Violation with Card and Button



Use Case 8 – Workflow Approval via WebEx Cards and Buttons

8.1. APPM SERVICE

1. APPM Service will constantly check database for events. Once an event matched the criteria it will send a message to pre-created Webex Space via code snippet below

```
curl --request POST --url https://webexapis.com/v1/messages --header 'Authorization: Bearer <ACCESS_TOKEN>' --header 'Content-Type: application/json' -d @<FORMATTED_MESSAGE>
```

Key	Value	Remarks
ACCESS_TOKEN	<access token>	Token is generated when creating bot at https://developer.webex.com/
FORMATTED_MESSAGE	<pre-formatted message>	

HEADER

Key	Value	Remarks
Content-Type	application/json	

EXAMPLE

SeaHackaton 26/8/2021, 6:29 PM

APPM Request Notification

Approval Password Request

Requestor: lukechen
 Start Date: 2021-08-26 18:29:00
 End Date: 2021-08-26 20:28:59
 Host name: OracleLinux_SystemB
 IP: 172.16.200.53
 Account: root
 Reason: System Error Checking

A request to access server. Please approve using action button.

Approve Reject More...

SeaHackaton 26/8/2021, 6:30 PM
 Request was approved by Luke

Use Case 9 – Approved Notification via WebEx Team Space

9.1. APPM SERVICE

- Once the request is approved, a direct message will be sent to the requester.

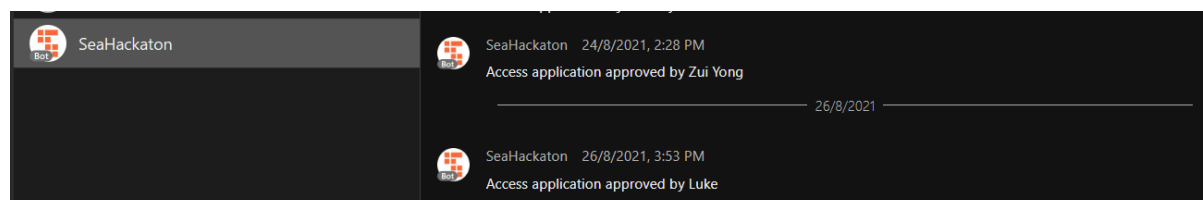
```
curl --request POST --url https://webexapis.com/v1/messages --header 'Authorization: Bearer <ACCESS_TOKEN>' --header 'Content-Type: application/json' -d '{"toPersonEmail": "<EMAIL_ADDR>", "text": "Access application rejected by <APPROVER>"}
```

Key	Value	Remarks
ACCESS_TOKEN	<access token>	Token is generated when creating bot at https://developer.webex.com/
toPersonEmail	<EMAIL_ADDR>	Room id retrieved via https://webexapis.com/v1/rooms
text	FORMATTED_MESSAGE	Access application rejected by <APPROVER_NAME>

HEADER

Key	Value	Remarks
Content-Type	application/json	

Example



END OF DOCUMENT